COEN-352 Tutorial #3

Abstract Data Types

ADT is a type of objects whose behaviour is defined by the set of values and operations.

- **ADT** vs **In-built data types**

- **Only Definition:** ADTs are implementation independent

- **Examples:** List, Stack, Queue…etc.

- **Case:** Implementation of Queue ADT based on **LL** vs **Arrays**

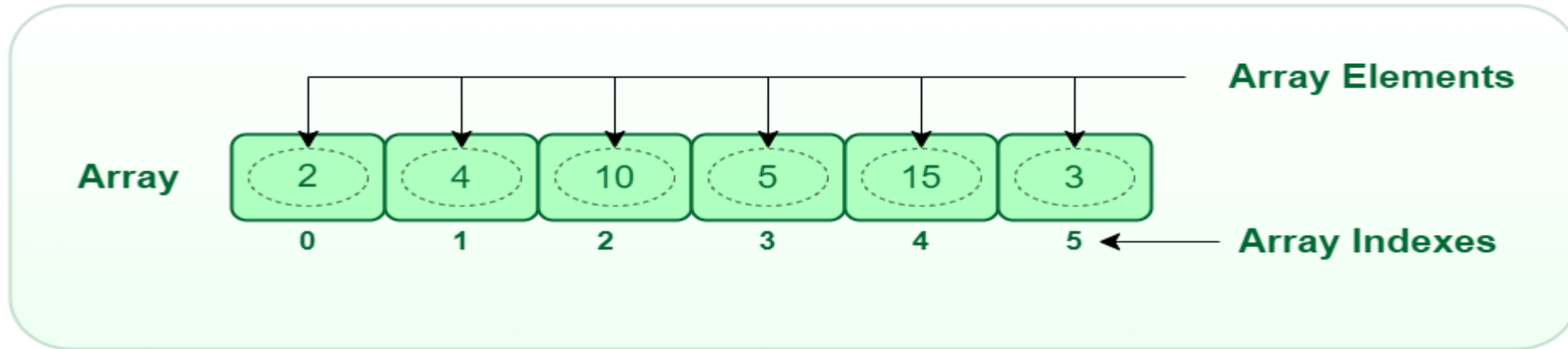- https://www.geeksforgeeks.org/abstract-data-types/

Some ADTs

ARRAYS:
- Elements are stored in sequential memory addresses, thus INDEXES
- Faster access to Element
- Really good in Arithmetic operations
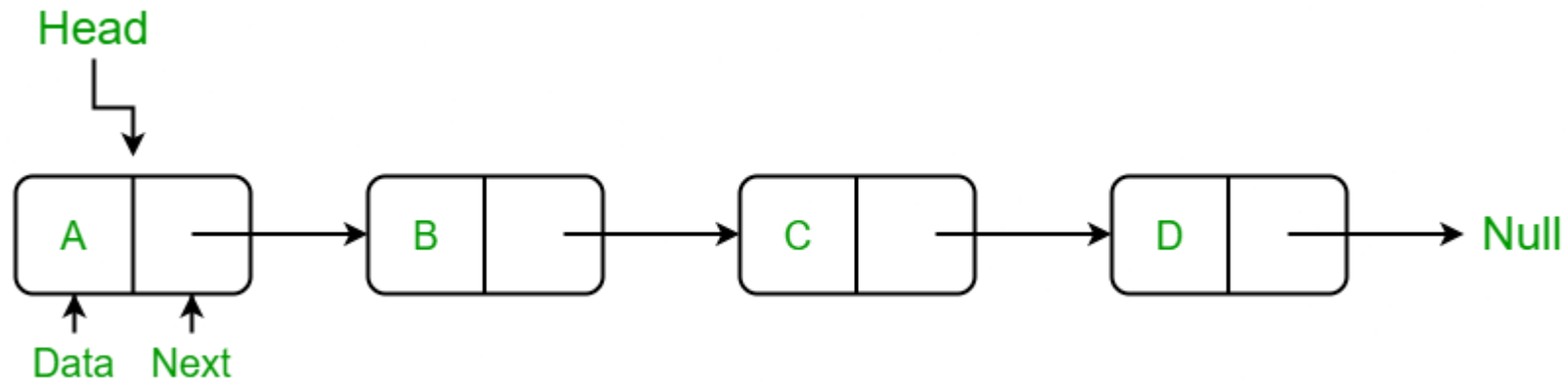- Defined Size, allocated during compile time

LISTS:
- Elements are not contiguous stored, they point to the next elements
- Elements are usually composed of pointers, one to the next and the other to a DS
- Really good with Insertions
- Dynamic Size, allocated during the run time

Arrays & Linked Lists

ARRAYS: https://www.geeksforgeeks.org/array-data-structure/

LINKED LISTS: https://www.geeksforgeeks.org/what-is-linked-list/
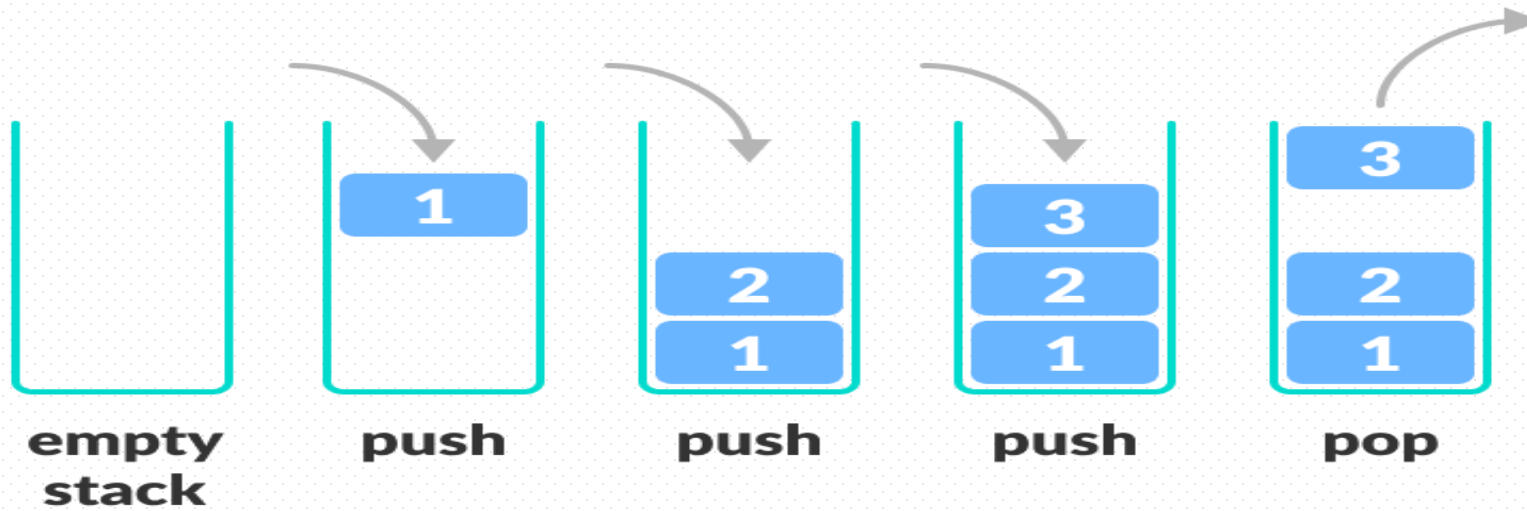
Some ADTs (cont'd)

STACKS:
- Elements are stored on top of each other
- It follows the LIFO (Last In, First Out) ordering
- Really good for recursion, hence it is used by the OS for return calls
- Can be based on an array or linked list
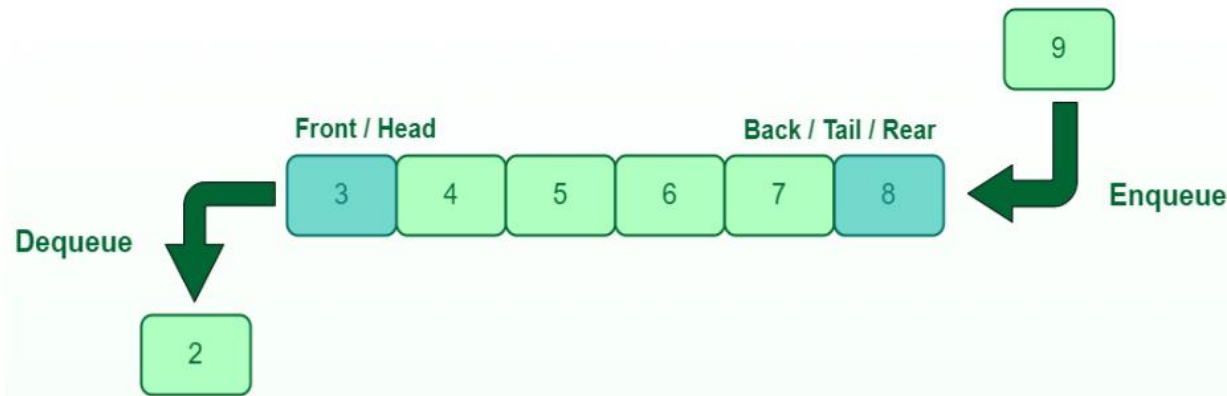- Operations: push(), pop(), peek(), size(), isEmpty(), isFull()

QUEUES:
- Elements are stored similar to the Stack DS
- It follows the FIFO (First In, First Out) ordering – open at both ends
- Really good with first come first served cases
- Can be based on an array or linked list
- Operation: enqueue(), Dequeue(), peek(), size(), isEmpty(), isFull()
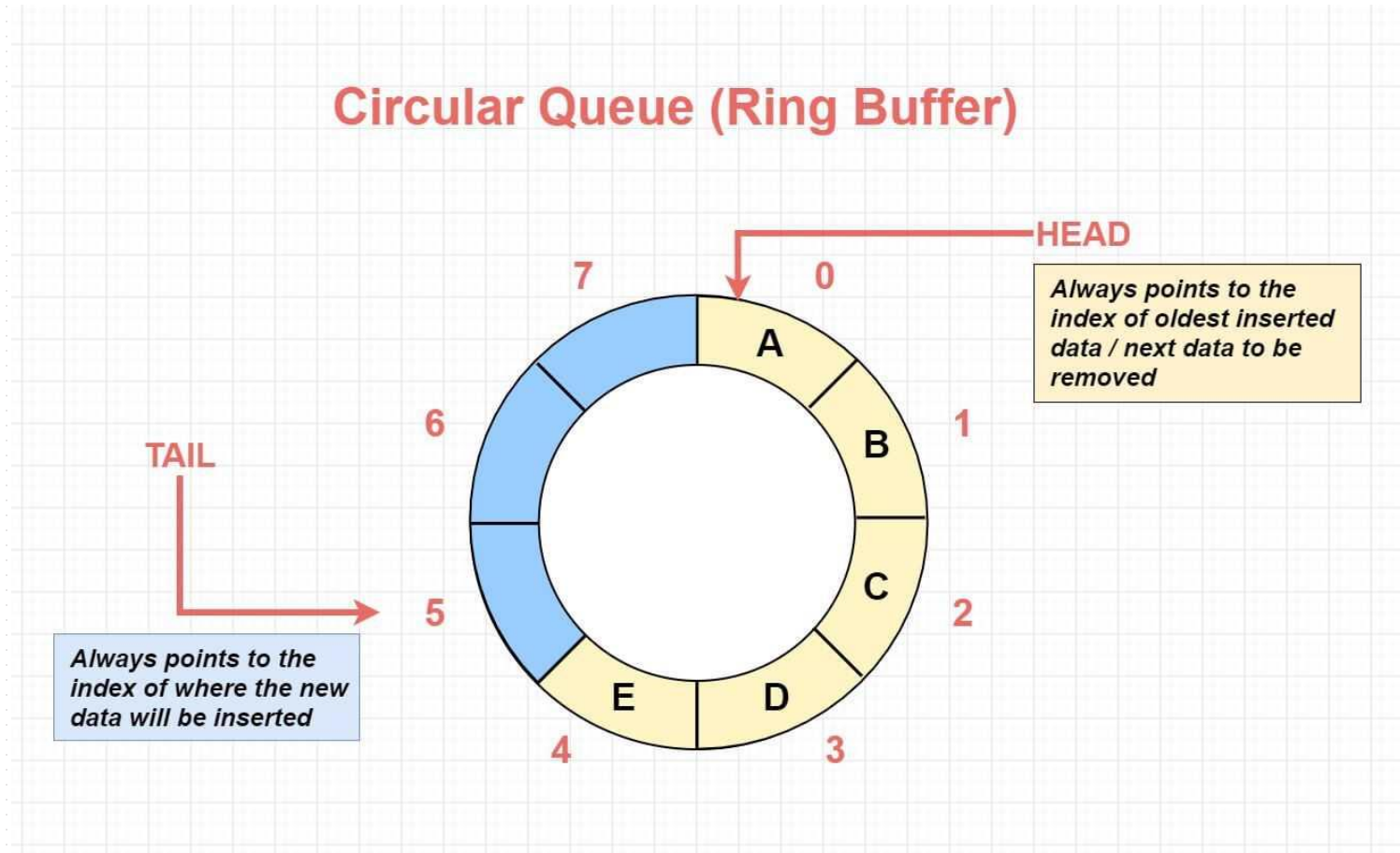
STACKS & QUEUES

STACK: https://www.programiz.com/dsa/stack

QUEUE: https://www.geeksforgeeks.org/queue-data-structure/

Circular QUEUE

Circular Queue (Ring Buffer)

HEAD
Always points to the index of oldest inserted data / next data to be removed

TAIL
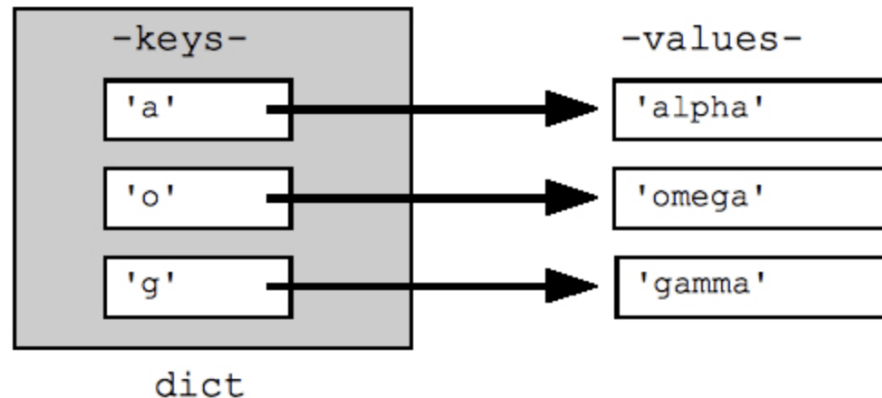*Always points to the index of where the new data will be inserted*

CIRCULAR QUEUE: https://www.sahinarslan.tech/posts/deep-dive-into-data-structures-using-javascript-circular-queue-ring-buffer

Some ADTs (cont'd)

DICTIONARIES:

- Similar to arrays but elements are stored based on a key
- Given a key, some value or data is returned – keys are associated with a value
- Can be though of as mapping
- Can be based on an array or linked list
- Operations: search(), insert(), delete(), isEmpty(), isFull()

DICTIONARIES: https://www.datacamp.com/tutorial/python-dictionaries

OTHER ADTs

- **BAGS**
- **TREE**
- **GRAPH**
- **HASH TABLE/MAP**
- TABLE
- VECTOR
- MAP
- MATRICES

THANK YOU